Multi-resolution image parametrization for improving texture classification

Luka Šajn[1], Igor Kononenko[1]

[1]University of Ljubljana
Faculty of Computer and Information Science
Tržaška 25, SI-1001 Ljubljana, Slovenia
{luka.sajn, igor.kononenko}@fri.uni-lj.si

November 20, 2007

Author responsible for correspondence: Luka Šajn[1], phone +386 41 795 980

Abstract

In the paper an innovative alternative to automatic image parametrization on multiple resolutions, based on texture description with specialized association rules, and image evaluation with machine learning methods is presented. The algorithm ArTex for parameterizing textures with association rules belonging to structural parametrization algorithms was developed. In order to improve the classification accuracy a multi-resolution approach is used. The algorithm ARes for finding more informative resolutions based on the SIFT algorithm is described. The presented algorithms are evaluated on several public domains and the results are compared to other well-known parametrization algorithms belonging to statistical and spectral parametrization algorithms. Significant improvement of classification results was observed when combining parametrization attributes at several image resolutions for most parametrization algorithms.

Our results show that multi-resolution image parametrization should be considered when improvement of classification accuracy in textural domains is required. These resolutions have to be selected carefully and may depend on the domain itself.

Keywords: Multi-resolution parametrization, texture classification, association rules, image processing, machine learning

1 Introduction

Images in digital form are normally described with matrices which are spatially complex and yet do not offer features that could uniformly distinguish between their predefined classes. Determining image features that can satisfactorily discriminate observed image classes is a hard task for which different algorithms exist. They transform the image from the matrix form into a set of numeric or discrete features (parameters) that convey useful information for discrimination between classes.

The motivation for our work was to develop an algorithm that finds resolutions at which image parametrization algorithms achieve more informative attributes since we observed that using parametrization parameters at more resolutions improves the classification accuracy.

This paper presents the algorithm ARes for selecting the resolution set (ArTex with resolutions - ARes) which yields more informative parametrization attributes when combining the parameters from the proposed resolutions. ARes was designed especially for structural image parametrization algorithms. Specifically, we use the ArTex algorithm [1; 2; 3]. ArTex (association rules for textures - ArTex) describes textures with specialized association rules which can also be extended with different measures. The obtained texture parameters are subsequently used for image classification with machine learning methods [4]. The idea of the ARes algorithm derives from the SIFT (Scale Invariant Feature Transform) algorithm [5]. ArTex and ARes are independent of the used machine learning algorithm.

The paper is organized as follows: in the next section we present the algorithm ArTex for parameterizing textures with association rules belonging to structural parametrization algorithms. In the second part the statistically evaluated results on several public domains are presented. The results are compared to four other well known algorithms which belong to statistical and spectral parametrization algorithms.

2 The ArTeX algorithm

This section presents textural features which are based on association rules. This algorithm was developed by Bevk and Kononenko [1; 2; 3]. A texture representation is given, which is an appropriate formalism that allows straightforward application of association rules algorithms. This representation has several good properties like invariance to global lightness and invariance to rotation. Association rules capture structural and statistical information and are very conve-

nient to identify the structures that occur most frequently and have the highest discriminative power.

2.1 Related work

Researchers have tried to characterize textures in many different ways. Most texture features are based on structural, statistical or spectral properties of the image. Some methods use textural features that include several of these properties. Well-known statistical features are based on gray-level co-occurrence statistics [6] which is used in the Image Processor program [7]. Examples of structural aspects are features of Voronoi tesselation [8], representations using graphs [9], representations using grammars [10] and representations using association rules [11]. The spectral features are calculated in a space which is closely related to textural features, for example frequency and amplitude. The most frequently used space transformations are Fourier, Laws [12], Gabor [13], and wavelet transform.

2.2 Association rules

Association rules were introduced back in 1993 [14]. The following is a formal statement of the problem: Let $\mathcal{I} = \{i_1, i_2, ..., i_m\}$ be a set of literals, called items. Let $\mathcal{D}$ be a set of transactions, where each transaction T is a set of items such that $T \subseteq \mathcal{I}$. We say that a transaction T contains X, if $X \subseteq T$. An *association rule* is an implication of the form $X \Longrightarrow Y$, where $X \subset \mathcal{I}$, $Y \subset \mathcal{I}$ and $X \cap Y = 0$. The rule $X \Longrightarrow Y$ holds in the transaction set $\mathcal{D}$ with *confidence* c if $c\%$ of transactions in $\mathcal{D}$ that contain X also contain Y. The rule $X \Longrightarrow Y$ has *support* s in the transaction set $\mathcal{D}$ if $s\%$ of transactions in $\mathcal{D}$ contain $X \cup Y$. The problem of discovering association rules says: Find all association rules in transaction set $\mathcal{D}$ with confidence of at least *minconf* and support of at least *minsup*, where *minconf* and *minsup* represent the lower boundary for confidence and support of association rules.

2.3 Texture representation

The use of association rules for texture description was independently introduced by [11]. In the present paper a slightly different approach is presented, which uses a different texture representation and a different algorithm for association rules induction, which were developed before we became aware of the work by [11].

Association rules are most widely used for data mining of very large relational databases.

In this section we give a representation of texture, which is suitable for processing with the association rules algorithms. To apply the association rules algorithms on textures, one must first define the terms which are used for association rules in the context of textures.

Pixel $\vec{A}$ of a texture P is a vector $\vec{A} = (X, Y, I) \in P$, where X and Y represent the absolute coordinates and I represents the intensity of pixel A.

Root pixel $\vec{K}$ is the current pixel of a texture $\vec{K} = (X_K, Y_K, I_K)$.

R neighborhood $N_{R,\vec{K}}$ is a set of pixels which are located in the circular area of radius R with root pixel $\vec{K}$ at the center. Root pixel $\vec{K}$ itself is not a member of its neighborhood.

$$\begin{gathered} N_{R,\vec{K}} = \{(X, Y, I) \,|\delta \leq R\} \setminus \vec{K} \\ \delta = \left\lfloor \sqrt{(X_K - X)^2 + (Y_K - Y)^2} + 0.5 \right\rfloor \end{gathered} \tag{1}$$

Transaction $T_{R,\vec{K}}$ is a set of elements based on its corresponding neighborhood. The elements of transaction are represented with Euclidean distance and intensity difference from the root pixel.

$$\begin{gathered} T_{R,\vec{K}} = \left\{ (\delta, I_K - I) | \, (X, Y, I) \in N_{R,\vec{K}} \right\} \\ \delta = \left\lfloor \sqrt{(X_K - X)^2 + (Y_K - Y)^2} + 0.5 \right\rfloor \end{gathered} \tag{2}$$

Transaction element is a two-dimensional vector $(r, i) \in T_{R,\vec{K}}$, where the first component represents the Euclidean distance from the root pixel and the second component represents the intensity difference from the root pixel.

Association rule is composed of transaction elements; therefore it looks like this

$$(r_1, i_1) \wedge \ldots \wedge (r_m, i_m) \Longrightarrow (r_{m+1}, i_{m+1}) \wedge \ldots \wedge (r_{m+n}, i_{m+n})$$

Transaction set $D_{P,R}$ is composed of transactions, which are derived from all possible root pixels of a texture P at certain neighborhood size R.

$$D_{P,R} = \left\{ T_{R,\vec{K}} \,\middle|\, \forall \vec{K} : \vec{K} \in P \right\}$$

This representation of a texture replaces the exact information of location and intensity of the neighboring pixels with more indecisive information of the distance and the relative intensity of neighboring pixels. This description is also rotation invariant.

Figure 1 illustrates the association rule $(1,1) \wedge (2,10) \Longrightarrow (1,15) \wedge (3,5)$, which can be read as follows: if a pixel of intensity 1 is found at distance 1 and a pixel of intensity 10 is found at distance 2, then there is also a pixel of intensity 15 at distance 1 and a pixel of intensity 5 at distance 3.

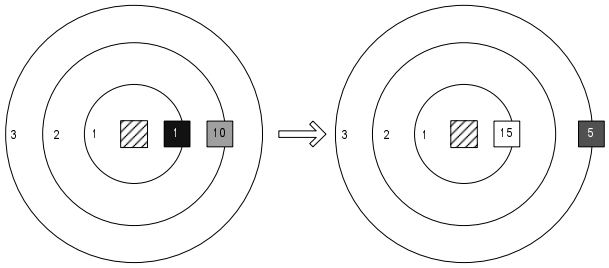


Figure 1: An illustration of association rule $(1,1) \wedge (2,10) \Longrightarrow (1,15) \wedge (3,5)$.

This representation is almost suitable for processing with general association rule algorithms. What is still to be considered, is the form of a transaction element. Association rule algorithms expect scalar values for transaction elements, whereas our representation produces a two-dimensional vector for a transaction element. Luckily, this issue can be easily solved. Let us say that the intensity of each texture point can have values within the interval $[0..(Q-1)]$ and that the neighborhood size is R. Take some transaction element (r,i), where i has a value from $[-(Q-1)..+(Q-1)]$ and r has a value from $[1..R]$. What is needed here is a bijective mapping that transforms each vector into its scalar representation. This can be achieved in many ways. A possible and quite straightforward solution is:

$$s = (2Q-1)(r-1) + i + (Q-1)$$

The transformation is also reversible:

$$r = 1 + s \; div \; (2Q-1)$$

$$i = s \; mod \; (2Q-1) - (Q-1)$$

Now it is possible to define a transaction that suits the general association rule algorithms:

$$T_{R,Q,\vec{K}} = \left\{ s \;\middle|\; \begin{array}{l} (r,i) \in T_{R,\vec{K}}, \\ s = (2Q-1)(r-1) + i + (Q-1) \end{array} \right\}$$

And finally we obtain the appropriate transaction set definition:

$$\mathcal{D}_{P,R,Q} = \left\{ T_{R,Q,\vec{K}} \;\middle|\; \forall \vec{K} : \vec{K} \in P \right\}$$

2.4 From association rules to feature description

Using association rules on textures will allow to extract a set of features (attributes) for a particular domain of textures. Algorithm ArTex is defined with the following steps:

- *Select a (small) subset of images F for feature extraction.* The subset F can be considerably small. Use at least one example of each typical image in the domain, i.e. at least one sample per class, or more if the class consists of subclasses.

- *Pre-processing of images in F.* Pre-processing involves the transformation of images to grey scale if necessary, the quantization of grey levels, and the selection of proper neighborhood size R. The initial number of grey levels per pixel is usually 256. The quantization process downscales it to say 32 levels per pixel. Typical neighborhood sizes are 3, 4, 5.

- *Generate association rules from images in F.* Because of the representation of texture, it is possible to use any algorithm for association rules extraction. We use *Apriori* and *GenRules* as described in [14].

- *Use generated association rules to extract a set of features.* There are two features associated with each association rule: support and confidence. Use these two attributes of all association rules to construct a feature set. The number of extracted features is twice the number of association rules (which could be quite numerous).

To clarify about statements, the formal algorithm is presented (see Algorithm 1). The algorithm takes five input parameters: a set of images I, neighborhood size R, texture quantization Q, minimum support *minsup* and minimum confidence *minconf*. Functions φ_{sup} and φ_{conf} are used to calculate support and confidence given an image and an association rule. The output of the algorithm is a feature set matrix d, where $d_{i,j}$ represents j-th feature of image i.

2.5 Extending the parameter set

In our model of texture the structure of the association rule also describes some aspects of the textural structure. Since we are interested in the parametric description of a texture, this structure has to be represented with one or more parameters. Till now we presented the basic algorithm which uses only basic interestingness measures, support and confidence, which were defined together with association rules [14]. They are still most widely used, but there are some concerns especially with confidence measure, which can be misleading in many practical

1: select F so that $F \subset I$
2: preprocess(F,R,Q)
3: **for all** $f \in F$ **do**
4: $\mathcal{D}$ = transactionModel(f,R,Q) STATE r_1 = apriori($\mathcal{D}$, minsup)
5: r_2 = genRules(r_1,$\mathcal{D}$, minconf)
6: $\rho_{sup} = \rho_{sup} \cup r_1$ {itemsets with support > minsup}
7: $\rho_{conf} = \rho_{conf} \cup r_2$ {rules with confidence > minconf}
8: **end for**
9: $i = 0$
10: **for all** $f \in (I \setminus F)$ **do**
11: $j = 0$
12: **for all** $\varrho \in \rho_{sup}$ **do**
13: $d_{i,j} = \varphi_{sup}(f, \varrho)$ {j-th attribute of i-th image}
14: $j = j + 1$
15: **end for**
16: **for all** $\varrho \in \rho_{conf}$ **do**
17: $d_{i,j} = \varphi_{conf}(f, \varrho)$
18: $j = j + 1$
19: **end for**
20: $i = i + 1$
21: **end for**
22: return d {d is a matrix of attribute values}

Algorithm 1: ArTex(images I, radius R, quantization Q, minsup, minconf)

situations, as shown by Brin et al. [15]. These authors also offered an alternative to evaluate association rules using χ^2 test. Unlike the confidence measure, χ^2 test could be used to find both positively and negatively correlated association patterns. However, the χ^2 test alone may not be the ultimate solution because the χ^2 test does not indicate the strength of correlation between items of association pattern. It only decides whether items of association pattern are independent of each other, thus it cannot be used for ranking purposes.

The χ^2 test was used to select interesting association patterns, which are later described by the Pearson's correlation coefficient (ϕ-coefficient) as advised in [16]. Besides, ArTex also uses an additional interestingness measure, which was selected via thorough experiments on various domains from a subset of collection made by Tan et al. [17]. From all tested measures the J-measure gave best results [2]: J-measure for rule $A \rightarrow B$ is defined as follows: $P(A,B)\log(\frac{P(B|A)}{P(B)}) + P(A,\overline{B})\log(\frac{P(\overline{B}|A)}{P(\overline{B})})$.

3 Multi-resolutional parametrization

Why use more resolutions? Digital images are stored in matrix form and algorithms for pattern parametrization basically use some relations between image pixels (usually first or second order statistics). By using only a single resolution, we may miss the big picture, and proverbially not see the forest because of the trees. Since it is computationally too complex to observe all possible relations between at least any two pixels in the image, we have to limit the search to some predefined neighborhood. These limitation makes relations vary considerably over different resolutions. This means that we may get completely different image parametrization attributes for the same image at different scales.

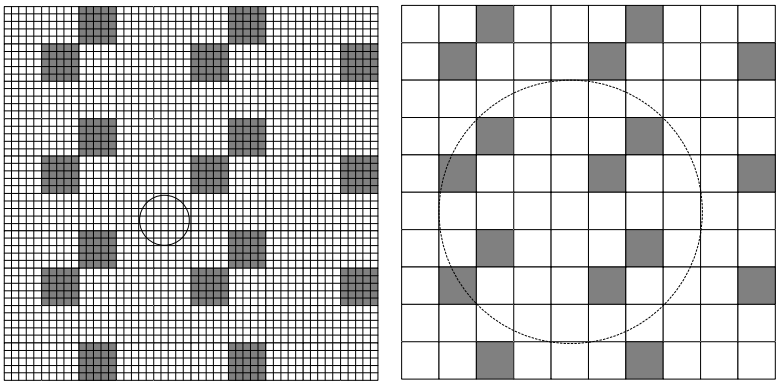
3.1 Parameters from many resolutions

In different existing multi-resolutional approaches [18; 19; 20] many authors are using only more resolutions, which are not determined on the basis of image contents. Usually two or three resolutions are used. Authors report better classification results when more resolutions are used and also observe that when using more than three resolutions, the classification accuracy starts to deteriorate. We have observed that in many cases authors use a set of resolutions by exponentially decreasing the resolution size ($\frac{100}{2^i}, i = 0..n-1$, where n represents the number of resolutions). However, we noticed that in many cases equidistant selection of resolutions ($\frac{100i}{n}, i = 1..n$ where n is the number of resolutions used) gives better results. Testing of our hypothesis is presented further in Section 5. When using exponential form of resolutions, a lower pattern content is examined and consecutively less significant attributes can be derived.

Another frequently used "multi-resolutional" approach is the wavelet transform [21], which describes textures with measures calculated by iterative image division. All the procedures mentioned above do not observe the contents of images.

Another extension of parameters for texture parametrization is derived from the issue of pattern's scale. Not every combination of scale and neighborhood size can guarantee that the pattern would be detected. The problem is illustrated in Figure 2.

To increase the possibility that the pattern will be detected, we propose a framework where the extraction of attributes is repeated at different texture resolutions and combined in one feature vector. The idea for the algorithm for automatic selection of a small subset of relevant resolutions is derived from the well-known SIFT algorithm [5]. Algorithm SIFT is designed as a stable local feature detector which is represented as a fundamental component of many image



(a) A very inadequate resolution (b) An adequate resolution

Figure 2: Detecting patterns at different scales

registration and object recognition algorithms.

3.2 Algorithm SIFT

Scale-invariant feature transform (or SIFT) [5] is one of several computer vision algorithms for extracting distinctive features from images. It can be used in algorithms for tasks like matching different views of an object or scene (e.g. for stereo vision) and Object recognition.

Features obtained by SIFT are invariant to image scale, rotation, and partially invariant (i.e. robust) to changing viewpoints, and change in illumination. The name Scale-invariant Feature Transform was chosen since the algorithm transforms image data into scale-invariant coordinates relative to local features. The feature representations found by SIFT are thought to be analogous to those of neurons in the inferior temporal cortex, a region used for object recognition in primate vision.

Due to the above mentioned properties (invariance and robustness), SIFT is an often used detection/description scheme. As a consequence, there are other algorithms which either emulate SIFT's functionality or try to outperform SIFT.

3.2.1 SIFT overview

First, the original image is progressively Gaussian blurred in a band from 1 to 2 resulting in a series of Gaussian blurred images (a scale-space produced by cascade filtering). Then these images are subtracted from their direct neighbors to produce a new series of images (with difference of Gaussian which approximates the Laplacian of the Gaussian).

The major steps in the computation of the image features are:

- **Scale-space extrema detection** - a specific type of blob detection where each pixel in the images is compared to its 8 neighbors and the 9 pixels (corresponding pixel+8 neighbors) of the other pictures in the series.
- **keypoint localization** - keypoints are chosen from extrema in the scale space.
- **orientation assignment** - for each keypoint, in a 16×16 window, histograms of gradient directions are computed (using bilinear interpolation).
- **keypoint descriptor** - representation in a 128-dimensional vector.

For the application of SIFT keypoints in matching and object recognition, Lowe [5] was applying a nearest neighbor algorithm, followed by a Hough transform [22] for object recognition.

3.3 ARes - SIFT modification for detecting informative resolutions

Since we are not interested in detecting stable image key-points but rather in detecting resolutions at which the observed image has most extremes, we have devised a new algorithm ARes (see Algorithm 2) for determining the resolutions for which more informative features can be obtained. The algorithm was designed especially for the ArTex parametrization algorithm (see Algorithm 1) but usually improves also the results with other parametrization algorithms as can be seen in the results section (see Section 5).

The ARes algorithm consequently resizes the image from 100% down to a predefined lowest threshold at some fixed step when detecting the appropriate resolutions. Both the lowest threshold and the resolution step are determined using the observed image data set. At each resize step the peaks are counted. Peaks are represented by pixels which differ from their neighborhood either as highest or lowest intensity. This algorithm can be implemented also with DOG (Difference-Of-Gaussian) [5] method which improves the time complexity with lower numbers of actual resizes required to search the entire resolution space.

The detected peak counts are recorded over all resolutions as a histogram. From the histogram the best resolutions are detected as the highest counts. The number of resolutions we want to use in our parametrization is predefined by the user. When there are several equal counts we chose as diverse resolutions as possible [23].

To demonstrate the difference between different resolution selections a sample generic image is given. Figure 3 demonstrates the use of classical equidistant resolutions (four are used) whereas Figure 4 shows the resolutions that ARes proposes. ARes finds only two resolutions

Require: set of input images Θ with known classes, number of desired resolutions η, number of images to inspect in each class γ, radius ϕ which the parametrization algorithm uses later on in the process

Ensure: subset of resolutions Π

1: $$W_{max} = \max_{i=1}^{|\Theta|}(\Theta_{i(width)}), H_{max} = \max_{i=1}^{|\Theta|}(\Theta_{i(height)})$$
{find the biggest image height and width}

2: extend the image sizes $\Theta_i \in \Theta$ to $W_{max} \times H_{max}$ with adding a frame of intensity equal to the average intensity of the original image Θ_i. New resized images are saved in the set Θ' {image sizes must be unified in order to be able to compare resolutions over different images}

3: $\delta = \frac{2*\phi}{3} \cdot \frac{1}{\min\{W_{max}, H_{max}\}}$ {set the resize step}

4: for each class add γ randomly selected images from the set Θ' into the set Θ_1

5: $\Omega = \{\}$

6: **for** $(\forall\theta \in \Theta_1)$ **do**

7: $\nu = 1.0$ {start with 100% resolution}

8: **while** $(\min\{W_{max}, H_{max}\} \cdot \nu > 3 \cdot \phi)$ **do**

9: $\theta_1 = resize(\theta, \nu)$ {change the observed image's size}

10: find local peaks in θ_1 with comparing each pixel's neighborhood inside $[3 \times 3]$ window

11: add the pair {ν, number of peaks} into the set Ω

12: $\nu = \nu - \delta$

13: **end while**

14: **end for**

15: order the set Ω by the number of descending peaks and resolutions

16: add first η resolutions from the ordered set Ω into the final set Π

Algorithm 2: Algorithm ARes for detecting a small subset of relevant resolutions

for the given example since they are sufficient for description of the objects present in the image. The larger picture is proposed since two elements disappear at lower resolutions and the smaller picture is proposed because it encounters the most extremes which consequently compactly describe the whole image. This appeared to be important when using structural parametrization algorithms since they search inside a fixed size pixel neighborhood. In the presented case SIFT chooses eight resolutions (40%, 38%, 41%, 30%, 39%, 29%, 42% and 8%) which are close to resolutions obtained with ARes algorithm but are too numerous, and the smallest and the most important resolution (8%) comes in as the eighth resolution, which means that we would have to parameterize textures at eight resolutions which is computationally much too complex.

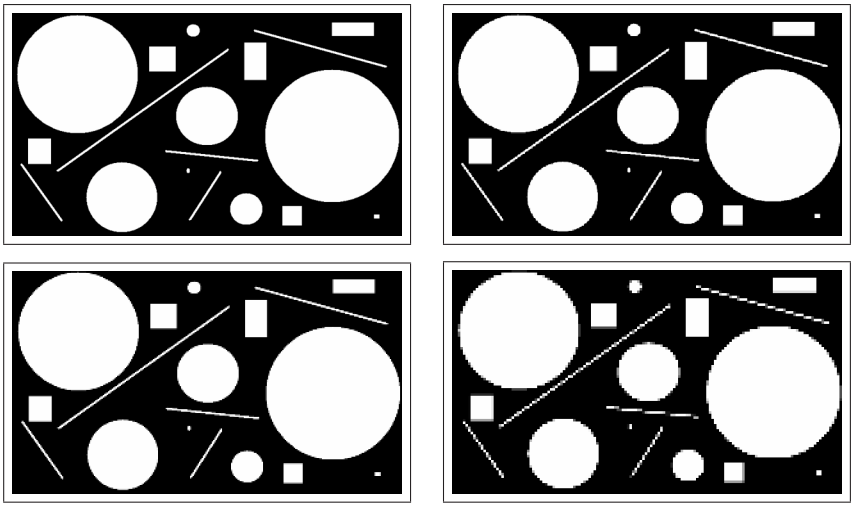


Figure 3: Equidistant resolutions ($n = 4$; 100.0%, 75.0%, 50.0%, 25.0%)

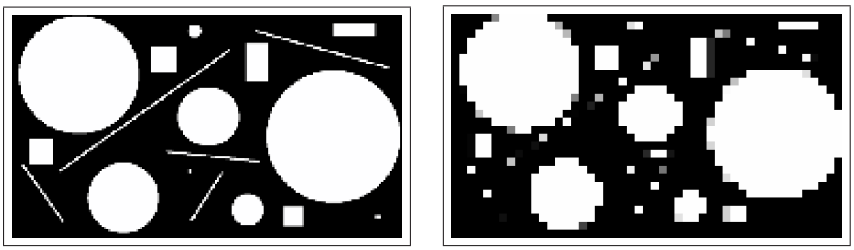


Figure 4: Resolutions detected with ARes (30.0% and 5.0%)

4 Testing the multi-resolution parametrization

The results of our parametrization with the ArTex algorithm are compared with the results of four other image parametrization algorithms (Haar wavelets [21], Laws filters [24], Gabor filters [25], Image processor [7] (implements many parameters of the first and the second order statistics [6]), and Laws' texture measures [24]).

Although algorithm ARes is designed for structural parametrization algorithms, we have checked its efficiency also for statistical ("Image Processor" IP) and spectral algorithms (Haar wavelets, Gabor and Laws filters).

4.1 Generic texture domains

Texture domains were chosen from public repositories which are usually used for evaluating the texture parametrization algorithms.

The following texture collections were used:

- Outex [26]:

  extensive Outex texture imageset developed at Finnish University Oulu consists of texture collections suitable for classification, segmentation and inquiry of pictorial contents.

  Domains for the presented experiments were taken from the *Classification* collections:

  - *Outex 0*

    Consists of 480 textures belonging to 24 different classes. All classes are equally distributed which means 20 textures per class. All textures are of the same size 128×128 and no intentional artifacts were added.

  - *Outex 1*

    Similar to *Outex 0* but 2112 textures are presented (88 textures per class) with smaller size 64×64.

  - *Outex 2*

    Again similar to *Outex 0* but 8832 textures are presented (368 textures per class) with even smaller resolution 32×32.

  - *Outex 10*

    Consists of 4320 textures belonging to 24 different classes (180 textures per class). Images are rotated using nine different rotations 0°, 5°, 10°, 15°, 30°, 45°, 60°, 75°, 90°. Texture sizes are 128×128.

  - *Outex 11*

    Consists of 960 textures belonging to 24 different classes (40 textures per class). Images are taken at two different resolutions 100dpi and 120dpi. Texture sizes are 128×128.

  - *Outex 12*

    Consists of 4800 textures belonging to 24 different classes (200 textures per class). Images are taken at different illumination and are rotated using nine different rotations 0°, 5°, 10°, 15°, 30°, 45°, 60°, 75°, 90°. Texture sizes are 128×128.

- Brodatz A [27]:

  Brodatz A collection is a subset of the Brodatz album [28] depicting different materials. The collection consists of 2048 textures belonging to 32 classes (64 textures per class). Each class is represented with 16 "original" surface images, 16 randomly rotated "original"

images, 16 cases of different resolutions of "original" textures and 16 randomly rotated and scaled "original" textures. Texture sizes are 64×64.

- Brodatz B [29]:
  Brodatz B collection is a subset of the above Brodatz album depicting different materials. The collection comprises of 1248 textures belonging to 13 classes (96 textures per class). Images are rotated using six different rotations 0°, 30°, 60°, 90°, 120° and 150°. Texture sizes are 128×128.

- Brodatz C [30]:
  Brodatz C collection is a subset of the above Brodatz album depicting different materials. The collection comprises of 6720 textures belonging to 15 classes (448 textures per class). Images are rotated using seven different rotations 0°, 30°, 60°, 90°, 120°, 150° and 200°. Texture sizes are 32×32.

It should be mentioned that the proposed algorithms were also used in some real medical domains [23; 31] where they have achieved significant results.

5 Results

The results obtained with all nine generic domains are presented. Statistical tests showing the successfulness of ARes algorithm are shown. In all tables the results of the basic resolution, N equally distributed resolutions and resolutions obtained by ARes and SIFT algorithms are presented. Tests were performed using four ($N = 4$) and eight ($N = 8$) resolutions. For testing the classification accuracy, the learning algorithm SVM [32] implemented in Weka [33] (SMO algorithm) was used. Tests were performed using 10-fold cross validation. To obtain features of individual class three randomly selected images were chosen. For algorithm ArTex the extension with J-measure was used. This approach gave better results on the applied generic domains which is also in agreement with results obtained by Bevk [2]. The selected radius was 5 pixels, all textures were quantified to 32 gray levels. When tests were performed, no feature subset selection was used to prevent the influence on results, since several algorithms which require different approaches for attribute selection were used. Typically for ArTex algorithm many attributes are obtained which cannot be selected in limited number as it is the case with other algorithms. The numbers of attributes for different algorithms are presented in Table 1. For

Haar and ArTex algorithms the result depends on the selected domain and resolution, therefore only an interval is presented.

Table 1: Typical numbers of attributes obtained with the presented algorithms

algorithm	number of attributes
ArTex	280-450
Gabor	48
Haar	20-30
Laws	26
IP	23

The results for all algorithms are presented in Tables 2-7. The comparison of classification accuracies of all five algorithms is shown in Table 8. Statistical evaluation of classification accuracies is given in Table 9. The abbreviations in tables have the following meanings: 100% res. represents the use of only the basic 100% resolution, eq. 4 and eq. 8 present the application of equidistant 4 and 8 resolutions respectively, ARes 4 and 8 present the use of 4 and 8 resolutions proposed by ARes, SIFT 4 and 8 present the use of 4 and 8 resolutions which are proposed by SIFT. In tables, missing results of SIFT algorithm are presented (marked by asterisk) which means that algorithm SIFT proposed too low resolutions which cannot be used by the parameterization algorithms. In such cases instead of results obtained by SIFT algorithm the results of the respective equidistant resolutions were used (e.g., instead of SIFT 8, eq. 8 was used). By such a procedure the comparison of averages can be achieved.

5.1 Results of ARes with ArTex

In Table 2 the results of ArTex algoritm for which ARes was developed are presented. Compared to other resolution selections the overall classification accuracy over all domains is significantly better when ARes was used, which proves its suitability. It is also evident that results obtained by SIFT algorithm are worse with respect to the results obtained by equidistant resolutions as well as resolutions proposed by ARes.

For additional testing of ARes algorithm the comparison of rank tests on the applied domains were performed. The comparison of results presented in Table 3 additionally confirms the effectiveness of ARes algorithm.

Table 2: ArTex classification accuracies over different resolution selections

domain	100% res.	eq. 4	ARes 4	SIFT 4	eq. 8	ARes 8	SIFT 8
outex 0	99.02	98.77	99.26	96.32	98.78	99.75	97.80
outex 1	95.20	94.85	96.42	82.45	96.13	97.21	87.50
outex 2	76.36	77.17	78.93	*77.17**	77.96	78.52	*77.96**
outex 10	99.08	99.60	99.62	98.78	99.27	99.72	99.11
outex 11	98.54	98.54	99.10	98.09	97.52	99.21	98.76
outex 12	98.77	99.37	99.45	98.92	99.26	99.64	99.26
brodatz A	83.71	95.03	97.13	93.65	95.59	97.44	92.93
brodatz B	99.42	99.75	99.92	99.34	99.59	99.92	99.50
brodatz C	55.12	56.82	58.79	30.92	57.53	57.26	*57.53**
average	89.47	91.10	92.07	86.18	91.29	92.07	90.04

Table 3: Rank tests of ArTex classification accuracies over different resolutions

domain	100% res.	eq. 4	ARes 4	*SIFT 4**	eq. 8	ARes 8	*SIFT 8**
brodatz A	7	4	2	5	3	1	6
brodatz B	6	3	1.5	7	4	1.5	5
brodatz C	6	5	1	7	2.5	4	2.5
outex 0	3	5	2	7	4	1	6
outex 1	4	5	2	7	3	1	6
outex 10	6	3	2	7	4	1	5
outex 11	4.5	4.5	2	6	7	1	3
outex 12	7	3	2	6	4.5	1	4.5
outex 2	7	5.5	1	5.5	3.5	2	3.5
avg.rank	5.61	4.22	1.72	6.39	3.94	1.50	4.61

5.2 Results of ARes with other parametrization algorithms

In Table 4 results of Haar algorithm are presented. For some domains ARes improves the results, but in most cases there are not significant improvements. On the contrary, ARes deteriorates results for most domains compared to equidistant resolutions.

Table 4: Haar classification accuracies over different resolution selections

domain	100% res.	eq. 4	ARes 4	SIFT 4	eq. 8	ARes 8	SIFT 8
outex 0	94.38	97.50	95.83	94.58	97.50	96.46	95.21
outex 1	91.29	94.51	94.22	88.87	95.60	95.41	92.00
outex 2	75.01	79.99	81.86	69.54	82.96	82.56	72.66
outex 10	85.53	93.17	92.25	87.22	94.93	95.05	90.46
outex 11	93.96	97.19	97.50	95.83	98.33	98.44	96.25
outex 12	88.04	94.52	93.85	89.29	96.40	96.56	93.19
brodatz A	66.01	74.90	78.56	69.67	80.17	82.62	74.32
brodatz B	94.07	96.40	96.31	95.35	97.04	97.04	96.31
brodatz C	64.46	65.22	67.53	*65.22**	67.02	67.31	*67.02**
average	83.64	88.16	88.66	83.95	89.99	90.16	86.38

In Table 5 results of Laws algorithm are presented. ARes deteriorates results for most domains compared to equidistant resolutions when Laws algorithm is applied. But even in this case the results with ARes are better compared to those obtained by the basic resolution.

In Table 6 results obtained with Gabor algoritm are presented. Also in this case there is no significant improvement compared to equidistant resolution but as in the previous case a better classification accuracy is evident compared to the basic resolution.

In Table 7 results of Image Processor algorithm are presented. In nearly 50% of domains the results with ARes are worse when compared to the equidistant resolutions and, as in previous cases, better when compared to the basic resolution.

For easier comparison in Table 8 the averages of all classification accuracies are presented.

The significant differences of resolutions and algorithms are presented in Table 9.

Rank comparison of classification accuracies of other algorithms (Table 10) shows that ARes in most non-structural algorithms for texture parametrization does not improve results compared to equally distributed space of resolutions, but in all cases it improves results regarding the basic resolution and resolutions proposed by SIFT algorithm.

Table 5: Laws classification accuracies over different resolution selections

domain	100% res.	eq. 4	ARes 4	SIFT 4	eq. 8	ARes 8	SIFT 8
outex 0	93.54	100.00	97.71	99.58	99.79	99.58	99.58
outex 1	93.13	98.20	98.20	96.78	98.53	98.72	97.49
outex 2	88.53	93.55	92.56	91.09	93.98	93.37	92.17
outex 10	85.37	98.17	91.46	95.23	98.84	95.69	96.74
outex 11	94.79	99.48	98.54	99.79	100.00	99.79	99.79
outex 12	83.31	98.27	90.00	96.35	98.96	94.81	97.75
brodatz A	62.89	86.67	84.96	82.47	89.36	88.33	84.96
brodatz B	78.21	94.39	84.54	90.95	96.23	93.99	94.31
brodatz C	66.38	69.18	67.66	*69.18**	69.70	68.33	*69.70**
average	82.91	93.10	89.51	91.27	93.93	92.51	92.50

Table 6: Gabor classification accuracies over different resolution selections

domain	100% res.	eq. 4	ARes 4	SIFT 4	eq. 8	ARes 8	SIFT 8
outex 0	99.58	99.38	99.79	98.54	99.79	99.58	98.75
outex 1	98.53	98.72	98.91	98.06	98.72	98.72	97.78
outex 2	94.79	95.63	95.69	90.13	95.86	95.81	91.40
outex 10	98.06	99.51	99.26	99.24	99.72	99.56	99.49
outex 11	99.58	99.79	99.79	99.48	99.90	100.00	99.58
outex 12	97.44	99.65	99.38	99.33	99.75	99.71	99.60
brodatz A	88.72	89.40	90.43	81.40	90.23	90.82	81.20
brodatz B	91.74	96.64	95.19	94.71	97.91	97.67	96.64
brodatz C	71.73	74.73	74.23	*74.73**	75.13	74.51	*75.13**
average	93.35	94.83	94.74	92.85	95.22	95.15	93.29

ARes algorithm was also checked by Friedman and Bonferroni-Dunn tests. At the confidence level $\alpha < 0.05$ the Friedman test shows a significant difference in ranks and the Bonferroni-Dunn test shows that at $\alpha < 0.05$ the critical distance between ranks is 2.686 and at $\alpha < 0.10$ it is 2.438. It can be concluded that ARes algorithm in combination with ArTex at four resolutions (see Table 11) significantly improves results compared with other resolutions in nearly all cases. For eight resolutions (see Table 12) the improvement is indicated in all cases. For

Table 7: Image Processor classification accuracies over different resolution selections

domain	100% res.	eq. 4	ARes 4	SIFT 4	eq. 8	ARes 8	SIFT 8
outex 0	75.83	82.50	81.46	82.71	86.46	85.63	85.83
outex 1	75.43	84.99	84.00	78.60	87.36	87.21	82.62
outex 2	62.61	70.75	68.91	61.98	72.07	70.12	65.95
outex 10	86.41	94.21	90.69	90.44	95.95	93.50	93.08
outex 11	78.85	88.65	85.63	83.96	90.94	89.38	86.88
outex 12	80.17	90.88	85.50	86.94	93.56	89.96	89.69
brodatz A	19.97	57.72	51.91	39.60	65.34	61.38	53.52
brodatz B	96.96	98.88	98.00	97.76	99.12	98.96	99.04
brodatz C	52.40	63.08	63.44	*63.08**	64.87	64.20	*64.87**
average	69.85	81.29	78.84	76.12	83.96	82.26	80.16

Table 8: Comparison of all classification accuracies over different algorithms

algorithm	100% res.	eq. 4	ARes 4	*SIFT 4**	eq. 8	ARes 8	*SIFT 8**
ArTex	89.47	91.10	92.07	86.18	91.29	92.07	90.04
Gabor	93.35	94.83	94.74	92.85	95.22	95.15	93.29
Haar	83.64	88.16	88.66	83.95	89.99	90.16	86.38
Laws	82.91	93.10	89.51	91.27	93.93	92.51	92.50
IP	69.85	81.29	78.84	76.12	83.96	82.26	80.16
average	83.84	89.70	88.76	86.07	90.88	90.43	88.47

other texture parametrization algorithms, ARes improves results only for the basic resolution and SIFT algorithm. Rank distances and statistical characteristics are presented in tables 11 and 12.

Table 9: Statistical evaluation of ARes for different resolutions and algorithms. The nonparametric Friedman test ($\alpha = 0,05$) was used. Numbers represent the number of domains where ARes achieves: "*significantly better / no significant difference / significantly worse*"

Alg. ARes in comb. with	number of resolutions n	eq. n	100% res.
ArTex	4	3/6/0	5/4/0
	8	5/4/0	6/3/0
Gabor	4	0/7/1	6/3/0
	8	0/9/0	7/1/0
Haar	4	3/5/1	8/1/0
	8	1/8/0	8/1/0
IP	4	0/3/6	9/0/0
	8	0/5/4	9/0/0
Laws	4	0/2/7	9/0/0
	8	0/5/4	9/0/0
sum		12/54/23	76/13/0

Table 10: Comparison of average classification accuracy ranks of different algorithms and resolutions

algorithm	100% res.	eq. 4	ARes 4	*SIFT 4**	eq. 8	ARes 8	*SIFT 8**
ArTex	5.61	4.22	1.72	6.39	3.94	1.50	4.61
Gabor	5.78	3.61	3.56	6.06	1.67	2.39	4.94
Haar	6.67	3.44	3.39	6.17	1.83	1.61	4.89
IP	6.89	3.28	4.89	5.72	1.06	2.67	3.50
Laws	7.00	2.67	5.33	4.72	1.28	3.44	3.56
avg.rank	6.39	3.44	3.78	5.81	1.96	2.32	4.30

Table 11: Distances between average classification accuracy ranks of different algorithms/resolutions when using ARes with four resolutions. (significant differences at $\alpha = 0.05$ are marked with ** and at $\alpha = 0.10$ with *)

algorithm	100% res.	eq. 4	SIFT 4	eq. 8	SIFT 8
ArTex	3.89**	2.50*	4.67**	2.22	2.89**
Gabor	2.22	0.06	2.50*	-1.89	1.39
Haar	3.28**	0.06	2.78**	-1.56	1.50
IP	2.00	-1.61	0.83	-3.83**	-1.39
Laws	1.67	-2.67*	-0.61	-4.06**	-1.78

Table 12: Distances between average classification accuracy ranks of different algorithms/resolutions when using ARes with eight resolutions. (significant differences at $\alpha = 0.05$ are marked with ** and at $\alpha = 0.10$ with *)

algoritem	100% res.	eq. 4	SIFT 4	eq. 8	SIFT 8
ArTex	4.11**	2.72**	4.89**	2.44*	3.11**
Gabor	3.39**	1.22	3.67**	-0.72	2.56*
Haar	5.06**	1.83	4.56**	0.22	3.28**
IP	4.22**	0.61	3.06**	-1.61	0.83
Laws	3.56**	-0.78	1.28	-2.17	0.11

5.3 Equidistant versus exponential resolutions

Here we are testing the hypothesis that equidistant resolutions ($\frac{100i}{n}, i = 1..n$ where n is the number of resolutions used) produce the same results as exponential resolution space ($\frac{100}{2^i}, i = 0..n-1$). We have observed significant differences in classification accuracies between equidistant and exponential resolution space when four ($N = 4$) or more resolutions were used. Results in Table 13 are presented only for five domains where textures are large enough to be resized with as small resolution as exponential space requires ($\frac{100}{2^{(4-1)}} = 12.5\%$, $128 \times 128 \rightarrow 16 \times 16$). Also in this case the Friedman nonparametric ($\alpha = 0,05$) test was used. In all domains except Brodatz B, equidistant resolution space (*eq. 4*) significantly outperforms the exponential one (*exp. 4*). Obtained results reject the tested hypothesis and support the claim that equidistant resolutions significantly outperform the exponential ones.

Table 13: Comparison of classification accuracies when using original, equidistant and exponential resolution space (*exp. 4* - exponential and *eq. 4* - equidistant resolution space) with ArTex.

Domain	100% res.	eq. 4	exp. 4
outex 0	99.02	98.77	97.30
outex 10	99.08	99.60	98.89
outex 11	98.54	98.54	96.17
outex 12	98.77	99.37	98.84
brodatz B	99.42	99.75	99.34
average	98.97	99.21	98.11

6 Conclusions

Significant improvement of classification results was observed when combining parametrization attributes at several image resolutions for most parametrization algorithms. In general, the main contribution of the study is the observation that equally-spaced multi-resolution performs better than other multi-resolution techniques.

In the case of ArTex algorithm our analysis of multi-resolution image parametrization showed certain relations between image content and its resolution with respect to the parametrization

quality. With all tested algorithms (structural, statistical and spectral) better results were obtained when more resolutions were used, which confirms our hypothesis that the choice of resolution space for parametrization of textures is to be considered. ARes algorithm was developed to support the resolution selection. In combination with ArTex it achieves significant improvement of classification results compared to a single resolution as well to equidistant resolution space. In addition, ARes also partially improves non-structural algorithm parameterizations.

Our results also indicate that equidistant resolution space performs better than exponential one, although many authors use the exponential resolution space by default. The exponential resolution space requires much bigger texture images when more than three resolutions are used.

The most significant ArTex improvement is achieved with the multi-resolution approach which is powered by the ARes algorithm designed especially for structural parametrization algorithms.

7 Future work

The presented algorithms open a whole new research area of multi-resolution image parametrization and enable many applications in medical, industrial and other domains where textures or texture-like surfaces are classified. The search of multi-resolution space can be achieved by many different algorithms we are currently working on. The hypothesis about equidistant resolutions being better than exponential is also being tested for other parametrization algorithms than ArTex and current results support this hypothesis.

References

[1] M. Bevk and I. Kononenko. Towards Symbolic Mining of Images with Association Rules: Preliminary Results on Textures. In Brito P. and Noirhomme-Fraiture M., editors, *ECML/PKDD 2004: proc. of the workshop W2 on symbolic and spatial data analysis: mining complex data structures*, pages 43–53, 2004.

[2] M. Bevk. *Derivation of texture features using association rules.* PhD thesis, Faculty of Computer and Information Science, University of Ljubljana, 2005. In Slovene.

[3] M. Bevk and I. Kononenko. Towards symbolic mining of images with association rules : preliminary results on textures. *Intelligent data analysis*, 10(4):379–393, 2006.

[4] I. Kononenko and M. Kukar. *Machine Learning and Data Mining: Introduction to Principles and Algorithms.* Horwood publ., 2007. (in press).

[5] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[6] R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(11):610–621, 1973.

[7] M. Bevk and I. Kononenko. A statistical approach to texture description of medical images: A preliminary study. In *The Nineteenth International Conference on Machine Learning ICML'02 Workshops*, Sydney, 2002.

[8] M. Tuceryan and A.K. Jain. Texture segmentation using voronoi polygons. *IEEE Transactions on Patern Analysis and Machine Intelligence*, pages 211–216, 1990.

[9] M. Tuceryan and A.K. Jain. *The Handbook of Pattern Recognition and Computer Vision (2nd Edition).* World Scientific Publishing Co., 1998.

[10] A. Rosenfeld and A.C. Kak. *Digital Picture Processing (2nd Edition)*, volume 2. Academic Press, Inc., 1982.

[11] J.A. Rushing, H.S. Ranganath, T.H. Hinke, and S.J. Graves. Using association rules as texture features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):845–858, 2001.

[12] K. I. Laws. *Textured image segmentation.* PhD thesis, Dept. Electrical Engineering, University of Southern California, 1980.

[13] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.

[14] R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 26-28 1993.

[15] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In Joan Peckham, editor, *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*, pages 265–276. ACM Press, 1997.

[16] P. Tan and V. Kumar. Interestingness measures for association patterns: A perspective. Technical Report TR00-036, Department of Computer Science, University of Minnesota, 2000.

[17] P. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the Eight ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[18] C. Bastos Rocha Ferreira and D. Leandro Borges. Automated mammogram classification using a multi-resolution pattern recognition approach. *SIBGRAPI01*, 00:76, 2001.

[19] M.L. Comer and E.J. Delp. Segmentation of textured images using a multiresolution gaussian autoregressive model. *Image Processing, IEEE Transactions on image processing*, 8(3):408–420, 3 1999.

[20] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[21] C.K. Chui. *An Introduction to Wavelets*. Academic Press, San Diego, 1992.

[22] P.V.C. Hough. Machine analysis of bubble chamber pictures. International Conference on High Energy Accelerators and Instrumentation, CERN, 1959.

[23] L. Šajn. *Multiresolution parameterization for texture classification and its application in analysis of scintigrafic images*. PhD thesis, Faculty of Computer and Information Science, University of Ljubljana, 2007. In Slovene.

[24] K. Laws. *Textured Image Segmentation*. PhD thesis, University of Southern California, 1 1980.

[25] S.E. Grigorescu, N. Petkov, and P. Kruizinga. Comparison of texture features based on gabor filters. *IEEE Trans. on Image Processing*, 11(10):1160–1167, 2002.

[26] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen. Outex- new framework for empirical evaluation of texture analysis algorithms. In *ICPR (1)*, pages 701–706, 2002.

[27] K. Valkealahti and E. Oja. Reduced multidimensional cooccurrence histograms in texture classification. *PAMI*, 20(1):90–94, 1 1998.

[28] P. Brodatz. *Textures - A Photographic Album for Artists and Designers.* Reinhold, Dover, New York, 1966.

[29] G.M. Haley and B.S. Manjunath. Rotation-invariant texture classification using a complete space-frequency model. *IEEE Tr. Im. Proc.*, 8(2):255–269, 2 1999.

[30] J.M. Carstensen. Cooccurrence feature performance in texture classification. In *The 8th Scandinavian Conference on Image Analysis, Tromsø, Norway*, pages 831–838, 5 1993.

[31] M. Kukar, L. Šajn, C. Grošelj, and J. Grošelj. Multi-resolution image parametrization in sequential diagnostics of coronary artery disease. In R. Bellazzi, A. Abu-Hanna, and J. Hunter, editors, *Artificial intelligence in medicine*, pages 119–129. Springer, 2007.

[32] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press, 2000.

[33] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, 1999.